

WHAT IS CLAIMED IS:

1. A method for maintaining dirty data comprising:
 - a. receiving a request to write data to a memory location that is cached to a cache;
 - b. writing the data to a plurality of cache lines in the cache, the plurality of cache lines written to being duplicate cache lines; and
 - c. marking the duplicate cache lines dirty.
2. The method of claim 1, wherein the cache is a multi-way set-associative cache.
3. A method for saving dirty data comprising:
 - a. reading data from a first dirty cache line in a plurality of cache lines in a cache, the first dirty cache line corresponding to a first memory location;
 - b. determining if the data is corrupt; and
 - c. if the data is not corrupt, then:
 - i. writing the data to the first memory location;
 - ii. marking the cache line available;
 - iii. determining at least one duplicate dirty cache line of the first dirty cache line; and

11 iv. marking each of the at least one duplicate dirty cache line as
12 an available cache line.

1 4. The method of claim 3, wherein each of the duplicate dirty cache lines is
2 marked invalid.

1 5. The method of claim 3, additionally comprising reading each of the
2 duplicate dirty cache lines and marking a duplicate cache line invalid if the
3 data is corrupt.

1 6. The method of claim 3, additionally comprising:

2 a. if the data on the first dirty cache line is corrupt, then marking the
3 cache line invalid, and determining if a duplicate cache line exists;

4 b. if a duplicate cache line exists, then determining if the data is
5 corrupt, and if the data is not corrupt, then:

6 i. writing the data to the first memory location; and

7 ii. marking the first dirty cache line available; and

8 c. if a duplicate cache line exists, and the data is corrupt, then
9 repeating the method at (a) until one of the following occurrences:

10 i. a duplicate cache line having data that is not corrupt is
11 found; and

12 ii. no more duplicate cache lines exist.

1 7. The method of claim 4, additionally comprising if a duplicate cache line is
2 not found, then terminating the method.

1 8. A method for saving dirty data comprising:

2 a. reading data from a first cache line in a plurality of cache lines in a
3 cache, the first cache line corresponding to a first memory location;

4 b. determining if the first cache line is a clean line;

5 c. determining if the data is corrupt; and

6 d. if the data is not corrupt, and the first cache line is not a clean line,
7 then:

8 i. writing the data to the first memory location;

9 ii. marking the first cache line available; and

10 iii. determining at least one duplicate dirty cache line of the first
11 dirty cache line and marking each of the at least one
12 duplicate dirty cache line invalid.

1 9. The method of claim 8, additionally comprising if the data is corrupt, and
2 the first cache line is not clean, then:

3 a. marking the cache line invalid, and determining if a duplicate cache
4 line exists;

5 b. if a duplicate cache line exists, then determining if the data is

6 corrupt, and if the data is not corrupt, then:

7 i. writing the data to the first memory location; and

8 ii. marking the first cache line available; and

9 c. if a duplicate cache line exists, and the data is corrupt, then
10 repeating the method starting at (a) until one of the following
11 occurrences:

12 i. a duplicate cache line having data that is not corrupt is
13 found; and

14 ii. no more duplicate cache lines exist.

1 10. The method of claim 9, additionally comprising if a duplicate cache line is
2 not found, then terminating the method.

1 11. The method of claim 10, additionally comprising if the data is corrupt, and
2 the first cache line is clean, then repeating the method starting at 8(a) until
3 no more associated cache lines exist.

1 12. A machine-readable medium having stored thereon data representing
2 sequences of instructions, the sequences of instructions which, when
3 executed by a processor, cause the processor to perform the following:
4 receive an instruction to write data to a memory location in a memory that
5 is cached to a cache;

6 write the data to a plurality of cache lines in the cache, the plurality of

7 cache lines written to being duplicate cache lines; and
8 mark the duplicate cache lines as dirty.

1 13. The machine-readable medium of claim 12, wherein the cache is a multi-
2 way set-associative cache.

1 14. A machine-readable medium having stored thereon data representing
2 sequences of instructions, the sequences of instructions which, when
3 executed by a processor, cause the processor to perform the following:
4 receive an instruction to write back at least one dirty cache line to a
5 memory;
6 select a first dirty cache line of a plurality of cache lines in a cache to write
7 back to the memory;
8 determine a corresponding memory location in the memory to write back
9 to by examining a tag portion of the first dirty cache line;
10 determine if the first dirty cache line has data that is corrupt;
11 writing the data back to the memory if the data is not corrupt, and marking
12 the first dirty cache line available; and
13 determining at least one duplicate dirty cache line of the first dirty cache
14 line and marking the at least one duplicate dirty cache line invalid.

1 15. The machine-readable medium of claim 14, additionally causing the
2 processor to perform the following:

3 if the first dirty cache line has data that is corrupt, then mark the first dirty
4 cache line as invalid, and determine if any duplicate dirty cache
5 lines of the first dirty cache line have data that is not corrupt;
6 if any one of the duplicate dirty cache lines has data that is not corrupt,
7 then:
8 write the data from the one duplicate dirty cache line to a
9 corresponding memory location;
10 mark the one duplicate dirty cache line available; and
11 mark the other duplicate dirty cache lines invalid.

1 16. The machine-readable medium of claim 15, wherein the cache is a multi-
2 way set-associative cache.

1 17. A machine-readable medium having stored thereon data representing
2 sequences of instructions, the sequences of instructions which, when
3 executed by a processor, cause the processor to perform the following:

- 4 a. receive an instruction to write back at least one cache line to a
5 memory;
- 6 b. select a cache line of a plurality of cache lines in an associated set
7 of a cache to write back to the memory;
- 8 c. determine if the cache line is a clean line;
- 9 d. determine a corresponding memory location in the memory to write

- 10 back to by examining a tag portion of the cache line;
- 11 e. determining if the cache line has data that is not corrupt;
- 12 f. if the data is not corrupt, then marking the cache line available, and:
- 13 i. writing the data back to the memory;
- 14 ii. determining if the cache line was a dirty cache line;
- 15 iii. if the cache line was a dirty cache line, then determining at
- 16 least one duplicate dirty cache line of the first dirty cache line
- 17 and marking the at least one duplicate dirty cache line
- 18 invalid; and
- 19 iv. if the cache line was not a dirty cache line, then repeating
- 20 the method at 17(b) until no more associated cache lines
- 21 exist.

- 1 18. The machine-readable medium of claim 17, additionally comprising if the
- 2 data is corrupt, then marking the line invalid, and:
- 3 a. if the cache line is a dirty cache line, determining at least one
- 4 duplicate dirty cache line of the first dirty cache line, and repeating
- 5 the method starting at 17(e) for a given one of the at least one dirty
- 6 cache line until one of the following occurs:
- 7 i. a duplicate dirty cache line having data that is not corrupt is
- 8 found; and

9 ii. no more duplicate cache lines exist; and

10 b. if the cache line is not a dirty cache line, then repeating the method
11 starting at 17(b).

1 19. The machine-readable medium of claim 18, wherein the cache is a multi-
2 way set-associative cache.

1 20. An apparatus comprising a cache controller to:

2 intercept a request to write data to a cached memory location;

3 determine an associated set having cache lines, the associated set

4 corresponding to the memory location; and

5 write the data to a plurality of cache lines in the associated set that are

6 available.

1 21. The apparatus of claim 20, further comprising invoking a replacement
2 policy of the cache to free up cache lines in the associated set if there are
3 no cache lines available.

1 22. A system comprising:

2 a main memory having at least one memory location that is cached to a
3 cache;

4 a cache having a plurality of cache lines to store data; and

5 a processor to receive an instruction to write data to a cached memory

6 location;
7 a cache controller to determine an associated set corresponding to the
8 memory location, the associated set having a plurality of cache
9 lines, and to write the data to a plurality of available cache lines in
10 the associated set.

1 23. The system of claim 22, the cache controller to additionally invoke a
2 replacement policy of the cache to free up cache lines in the associated
3 set if no available cache lines in the associated set are available.

1 24. A system comprising:
2 a main memory having at least one memory location that is cached to a
3 cache;
4 a cache having a plurality of cache lines to store data; and
5 a first means to receive an instruction to write data to a cached memory
6 location;
7 a second means to determine an associated set corresponding to the
8 memory location, the associated set having a plurality of cache
9 lines; and
10 a third means to write the data to a plurality of available cache lines in the
11 associated set.